Course Title:

System Analysis and Design (BIT 253)

BIT 4th semester-Tribhuvan University (TU)

Compiled by Ujjwal Rijal
rijalujjwal09@gmail.com

Unit-3

Analysis

Compiled by Ujjwal Rijal
rijalujjwal09@gmail.com

❑ Introduction

- Analysis is the first SDLC phase where we begin to understand the need for the system changes.
- The analysis team should not take the analysis process for granted or attempt to speed through it.
- Most observers would agree that many of the errors in developed systems are directly traceable to inadequate efforts in the analysis and design phases of the life cycle.
- ***The analysis process is generally divided into two main activities or sub-phases:***

✓ **Requirements Determination:** This is primarily a fact – finding activity.

✓ **Requirements Structuring:** This activity creates a thorough and clear description of current business operations and new information processing services. ***Requirements structuring can further be done in two ways i.e. structuring system process requirements*** and ***structuring system data requirements***.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Performing System Requirements Determination

- Once management has granted permission to pursue development of a new system and project is initiated and planned, we begin determining what the new system should do.
- All of the system requirements are carefully documented and prepared for further structuring.
- The primary deliverables from requirements determination are various forms of information gathered during the determination process: transcripts of interviews, notes from observations, analysis of documents, responses from questionnaires, sets of forms and reports, etc.
- ***These information could be grouped into the following three groups:***

✓ ***Information collected from conversations with or observations of users (interview transcripts, notes from observations, etc.).***

✓ ***Existing written information (business mission or strategy, forms, reports, training manuals, flow charts and documentation of existing system, etc.).***

✓ ***Computer-based information (results from JRP sessions, CASE repository contents, etc.).***

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- ***The various characteristics of a good system analyst possessed during the requirements determination sub – phase are as follows:***

✓ **Impartiality:** Our role is to find the best solution to the business problems or opportunities. We must consider the issues raised by all the parties and find the best organizational solution.

✓ **Impertinence:** We should ask question everything under here. We need to ask questions such as; Are all transactions processed in the same way? Could anyone be charged something other than the standard price?, etc.

✓ **Relax constraints:** Here, we assume that anything is possible and eliminate the infeasible requirements. For example, do not accept this statement: "We have always done it that way, so we have to continue the procedure". Traditions may not always be reasonable because they are different from the rules and policies.

✓ **Attention to details:** It states that every facts must fit with every other facts so that the system works properly. One element out of place means that even the best system will fail at the same time.

✓ **Reframing:** Analysis is a creative process, so we must set challenge for ourselves to look at the organization and its activities in new ways or new creative approach.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Traditional Methods for Determining Requirements

- At the core of system analysis, there is the collection of information. One of the best ways to gather information is to talk to the people directly or indirectly or to gather copies of documentation relevant to the current systems and business processes.
- ***The traditional methods for determining system requirements are as follows:***

I. Interviewing and Listening: Interviewing is the one of the primary ways of collecting information, analysts gather information about an IS project.

- Here, it says that individually interview people, informed about the operations and the issues of the current and future needs of the system.
- ***Experienced analysts commonly accept some of the following best practices:***

✓ Prepare the interview carefully, including appointment, priming question, checklist, agenda, and questions.

✓ Listen carefully and take note during the interview (tape record if possible).

✓ Review notes within 48 hours after the interview.

✓ Be neutral.

✓ Seek for diverse views, etc.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

II. Interview group of people: Here, it says to interview group of people with diverse needs to find the similarities and differences among the system requirements illustrated.

III. Questionnaires: Questionnaires have the advantage of gathering information from many people in a relatively short time and of being less biased in the interpretation of their results. Choosing the right questionnaires respondents and designing effective questionnaires are the critical issues in this information collection method. To conduct an effective survey, the analyst should group the users properly and design different questionnaires for different group.

IV. Observations: Observation is primarily useful for capturing what is already in existence and enables several other types of requirement tools, which is existing use – case scenario. Here, analysts ask the users what they like and what they don't, and requirement is specified which is required for the system build – up. Users can be observed directly as well, as they are not reliable in fetching proper communication.

Observation also says that observe workers at selected time to see how data are handled and what information people need to do their jobs.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

V. Analyzing procedures and other Documents: It says that study business documents to discover repeated issues, policies, rules, and directions as well as concrete examples of the use of data, information in the organization.

- ***In the documents, the document analyst can find information about the following illustrated criteria:***

✓ Problems with the existing system.

✓ Organizational direction that can influence information system requirements.

✓ Opportunities to meet the new needs if only certain information or information processing are available.

✓ Titles and names of key individuals who have an interest in the relevant existing systems.

✓ Values of the organization or individuals who can help determine priorities for different capabilities desired by different users.

✓ Data, rules for processing data, and principles by which the organization operates that must be enforced by the information system, etc.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Contemporary Methods for Determining Requirements

- Here, there are various techniques to collect information about the current system, the organization area requesting the new system and what the new system be like.
- ***The contemporary methods for determining system requirements can be divided into two parts: JAD and the prototyping model.***

I. Joint Application Design (JAD)

- The main idea behind JAD is to bring together the key users, managers, and system analysts involved in the analysis of a current system.
- In that respect, JAD is similar to a group interview; a JAD, however, follows a particular structure of roles and agendas i.e. quite different a group interview during which analyst controls the sequence of questions and answered by the users.
- In short, the primary purpose of using JAD concept in the analysis phase is to collet the system requirements simultaneously from the key people involved with the system design.
- JAD means bringing session users, sponsors, analysts and others together in a JAD session to discuss and review the system requirements.
- ***The typical participants in a JAD session are as follows:***

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- ✓ **JAD session leader:** This person has been involved in group management and facilitation as well as in the system analysis.
- ✓ **Users:** Users are the only one who have a mutual understanding of what it means to use the system on a daily basis.
- ✓ **Managers:** Managers of the work groups are those who use the system in question that provides insight into new organizational directions, motivations for and for requirements determination during the JAD session.
- ✓ **Sponsors:** As a major undertaking due to the project's expense, or JAD must be sponsored by someone at a relatively high level at the company.
- ✓ **System Analysts:** Analysts in the JAD session are to learn from users and managers, not to run or dominate the process.
- ✓ **Scribe:** Here, the scribe text notes during the JAD session. Notes can be taken on a laptop using a word processor, or notes and diagrams may be entered directly into a CASE tool.
- ✓ **IS staff:** Besides system analysts, other IS staffs such as programmers, planners, database analysts, and data centre personnel may attend to learn from the discussion and possibly contribute the current system under construction.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

II. Using the Prototyping model

- The prototyping model suggests that before developing the actual software, a working prototype of the system should be built. A prototype is a partially developed product which has limited functional capabilities, low reliability and inefficient performance.
- The model starts with an initial requirement gathering phase. A quick design is carried out and the prototype model is built using several shortcuts. The shortcuts might evolve using inefficient, inaccurate dummy systems.
- The prototype product usually turns out to be a very crude version of the actual system. A function may produce the desired result by using a table lookup rather than performing the actual computation.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

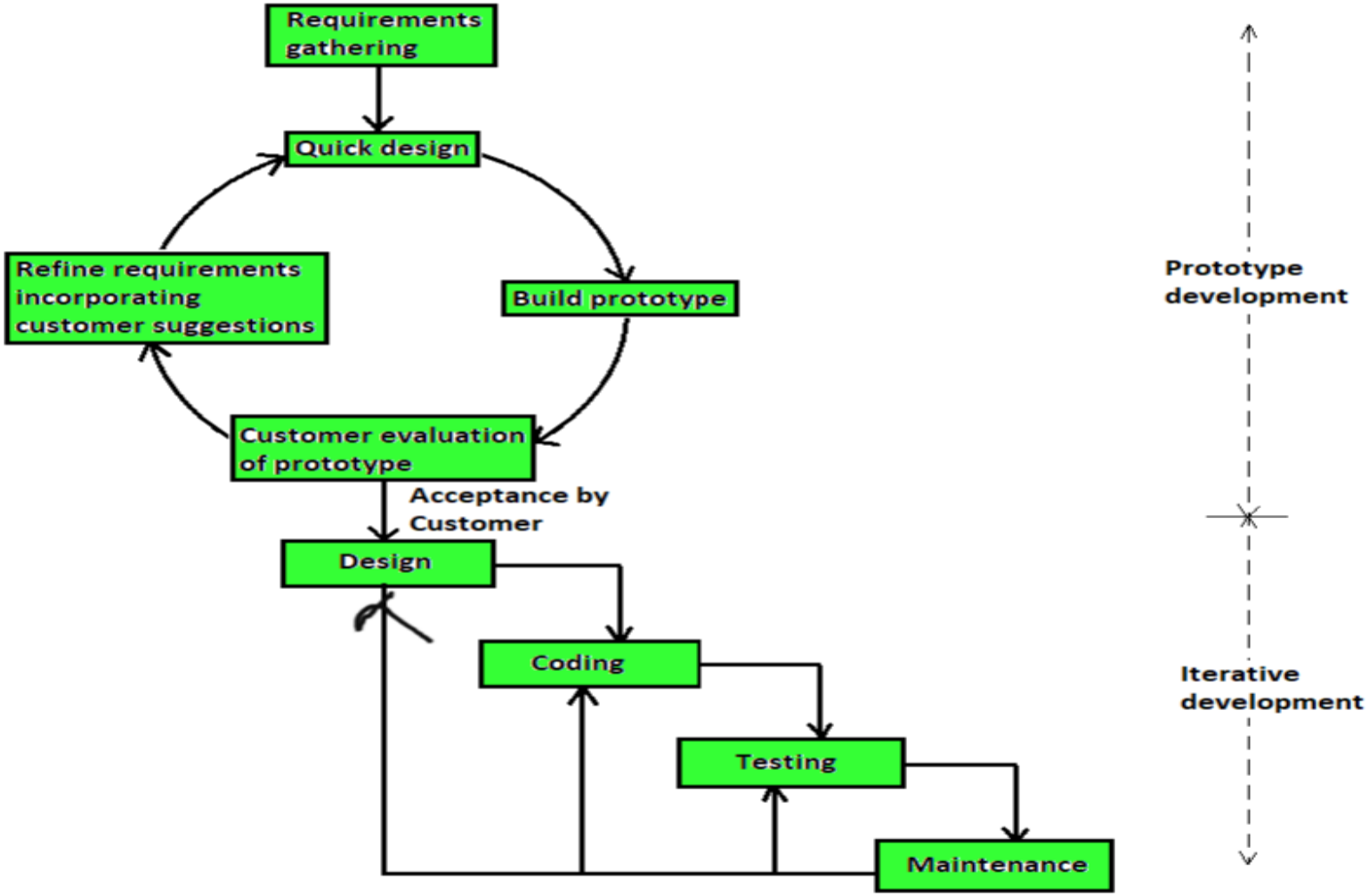


Fig:The Prototyping Model

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- Based on the user's feedback, requirements are refined. This cycle continues until the users approves the prototype. The actual system is then developed using the classical waterfall life cycle model.
- *There are usually two types of prototyping models:*

✓ **Evolutionary Development Prototype Model:** Objective of this model is to work with the users in order to explore their requirements and deliver a final system. It starts with the parts of the system that are understood and then evolved as the new system as proposed by the users.

✓ **Throwaway Prototype Model:** Objective of this model is to understand the user requirements and develop the better requirements definition of the system. It concentrates on the poorly understood components.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Radical Methods for Determining Requirements

- Analysts use system requirements determination to understand the current problems and support entities, as well as to determine what is needed and desired in future systems.
- The overall process by which current methods are replaced with radically new methods, is generally referred as **business process re-engineering (BPR)**.
- Even if the term **BPR** may seem dated to some, process orientation remains a testing legacy of the **BPR** movement.
- ***BPR consists of the following processes:***

✓ **Identifying processes to re – engineer.**

✓ **Disruptive Technologies.**

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- ✓ **Identifying processes to re – engineer:** The first step in any BPR effort relates to understand what processes to change. The key business processes are the structured, measured set of activities designed to produce a specific output for a particular customer or the target market.
- ✓ **Disruptive Technologies:** Once the key business processes and activities have been identified, information technologies must be applied to radically improve business processes.
- Disruptive technologies are those technologies that enable the breaking of long – held business rules that exhibit organization from making the radical business changes.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Structuring System Process Requirements

- Structuring system process requirements concentrates on the definition, structure and relationship with data and processes.
- The characteristics of data captured during process modeling are crucial in the design of databases, programs, computer skills, and printed reports. This information is essential in ensuring data integrity in an information system.
- Here, the focus will be on one tool that is used to represent the information gathered as a part of requirements determination – Data Flow Diagram (DFD).
- DFD also shows the processes that change or transform data because DFD concentrates on the movement of data between processes. These diagrams are called **process models**.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Process Modeling

- Process modeling involves graphically representing the functions or processes that are capture, manipulate, distribute, and store the data between a system and its environment, and between the components within a system.
- Moreover, a process model is a representation of reality that can be built for existing system as a way to better understand those systems or for proposed systems as a way to document business requirements or technical designs.
- **The process modeling determination is best depicted through the use of Data Flow Diagram (DFD).**

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Process Modeling: Data-Flow Diagram (DFD)

- DFD is a common form of a process model. In fact, DFD is a picture of the movement of data between external entities and the processes, and the data stores within the system.
- **DFD consists of the following four symbols or parts or components:**

I. Process

- Bubbles (or circles) or rectangles (or squares) are used to indicate where the incoming data flows are processed and then transformed into outgoing data flows.

II. Data Flow

- Arrows making the movement of data through the system indicate data-flows.
- Data-flow is a transfer of data between two entities through processes.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

III. External Entities

- These represent the source of data as input to the system. These are external to the system; we study these in terms of business processes.
- They are also the destination of the system data. They are represented by squares or rectangles.

IV. Data Stores

- A data store represents the storage of persistent data required and/or produced by the process.
- External entities outside the system can be called data stores.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

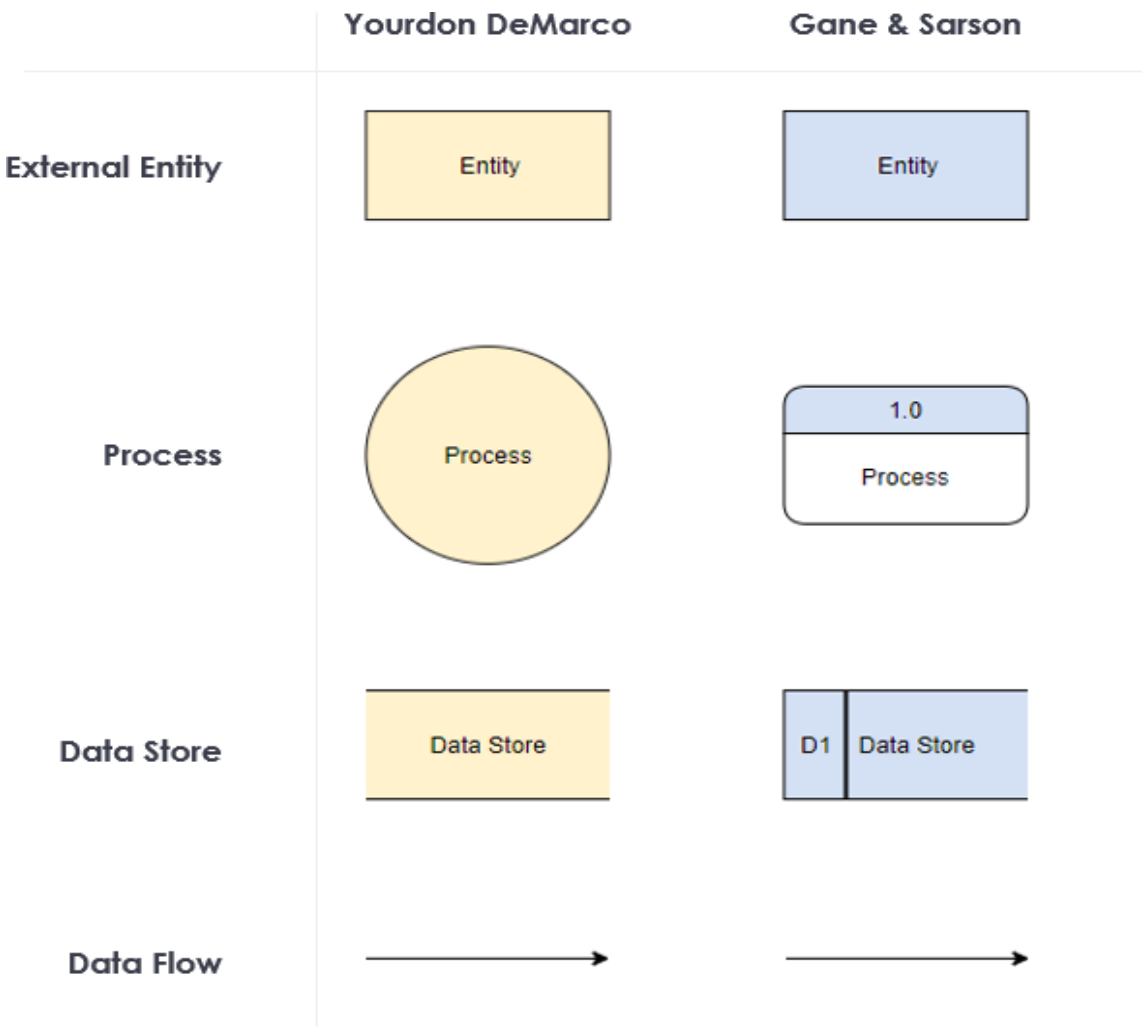


Fig: Symbols used in DFD

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Types of DFD

- **DFDs are categorized as either logical or physical.**

Logical DFD	Physical DFD
A type of DFD that depicts how the business operates	A type of DFD that depicts how the system is implemented
Focuses on the business activities	Focuses on the system implementation
A process is a business activity	A process is a software program or manual procedures
A data store is a collection of information	Data stores are databases, computer files and paper files
Simple	Complex

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- Logical DFDs – *Advantages*
  - Better communication with system users
  - Better stability for the system
  - Better business understanding for analysts
  - Better flexibility and maintenance

- Physical DFDs – Advantages
  - Easier to categorise processes as manual or automatic
  - Better description of processes
  - Better for ordering processes into a sequence
  - Better for imposing controls

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Levels of DFD

1. Level-0 DFD (Context Diagram)

- A level-0 DFD, also known as context diagram, shows a data system as a whole and emphasizes the way it interacts with external entities. It identifies the flow of information between the system and external entities.

2. Level-1 DFD

- A level-1 DFD presents a more detailed view of the system than context diagram, by showing the main sub-processes and stores of data that make up the system as a whole. In short, we can think of a level-1 DFD as an exploded view of the context diagram.

3. Level-2 DFD

- A level-2 DFD offers a detailed look at the processes that makeup an information system than a level-1 DFD does. It can be used to plan or record the specific makeup of a system. It is also known as lower level in the hierarchy of DFD levels.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ....

(For the examples of DFD construction, check out your note copy. All of these have been discussed and written during the class's session).

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Logic Modeling

- A logic model is a graphical depiction that represents the shared relationships among the resources, activities, outputs, outcomes, and impacts for your program.
- It depicts the relationship between your program's activities and its intended effects.
- Logic modeling can also be used to show when processes on a DFD occur.
- Logic modeling will be generic without taking syntax of a particular programming language.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ....

- Each processes on the lowest level DFD will be represented by one or more of the following:

1. Structured English
2. Data Dictionary
3. Pseudocodes
4. Decision Tables
5. Decision Trees
6. State-Transition Diagrams (Discussed)
7. Activity Diagrams (Discussed)
8. Sequence Diagrams (Discussed)

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Logic Modeling with Structured English

- Structured English is derived from structured programming language which gives more understandable and precise description of processes.
- It is based on procedural logic that uses construction and imperative sentences designed to perform operation for the desired action.
- It is best used when sequences and loops in a program must be considered and the problem needs sequences of actions with decisions.
- It does not have strict syntax rule. It expresses all logic in terms of sequential decision structures and iterations.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ....

- For example, consider the following sequence of actions:

```
if customer pays advance
   then
      Give 5% Discount
   else
      if purchase amount >=10,000
         then
            if  the customer is a regular customer
               then Give 5% Discount
            else  No Discount
         end if
      else No Discount
   end if
end if
```


Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Logic Modeling with Data Dictionary

- A data dictionary is a structured repository of data elements in the system.
- It stores the descriptions of all DFD data elements, i.e., details and definitions of data flows, data stores, data stored in data stores, and the processes.
- A data dictionary improves the communication between the analyst and the user.
- It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

- For example, refer the following table:

Sr.No.	Data Name	Description	No. of Characters
1	ISBN	ISBN Number	10
2	TITLE	title	60
3	SUB	Book Subjects	80
4	ANAME	Author Name	15

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Logic Modeling with Pseudo-Codes

- A pseudocode does not conform to any programming language and expresses logic in plain English.
- It may specify the physical programming logic without actual coding during and after the physical design.
- It is used in conjunction with structured programming.
- It replaces the flowcharts of a program.
- Pseudocode is simply an implementation of an algorithm in the form of annotations and informative text written in plain English.
- It has no syntax like any of the programming languages and thus cannot be compiled or interpreted by the computer.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

Problem :

Find an algorithm that can tell if a given number is even or odd.

Algorithm:

1) get a number
2) divide by 2
3)check the remainder if its 0 then answer is "Even" otherwise "odd"

Pseudocode:

```
INPUT Number
Remainder = Number % 2
IF Remainder = 0
  THEN PRINT Number,"
is even"
  ELSE PRINT Number,"
is odd"
ENDIF
```


Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Logic Modeling with Decision Tables

Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.

- It is useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.
- It is a matrix containing row or columns for defining a problem and the actions.

Components of a Decision Table

- **Condition Stub** – It is in the upper left quadrant which lists all the condition to be checked.
- **Action Stub** – It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.
- **Condition Entry** – It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.
- **Action Entry** – It is in lower right quadrant which indicates the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ...

❖ Procedures for constructing Decision Tables

- ✓ Name the conditions and the values that each condition can assume.
- ✓ Name all the possible actions that can occur.
- ✓ List all the possible rules ($2^{\text{no. of conditions}}$, in general).
- ✓ Define the action for each rule.
- ✓ Simplify the decision table.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

The entries in decision table are given by Decision Rules which define the relationships between combinations of conditions and courses of action. In rules section,

- Y shows the existence of a condition.
- N represents the condition, which is not satisfied.
- A blank - against action states it is to be ignored.
- X (or a check mark will do) against action states it is to be carried out.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

For example, refer the following table –

CONDITIONS	Rule 1	Rule 2	Rule 3	Rule 4
Advance payment made	Y	N	N	N
Purchase amount = Rs 10,000/-	-	Y	Y	N
Regular Customer	-	Y	N	-
ACTIONS				
Give 5% discount	X	X	-	-
Give no discount	-	-	X	X

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

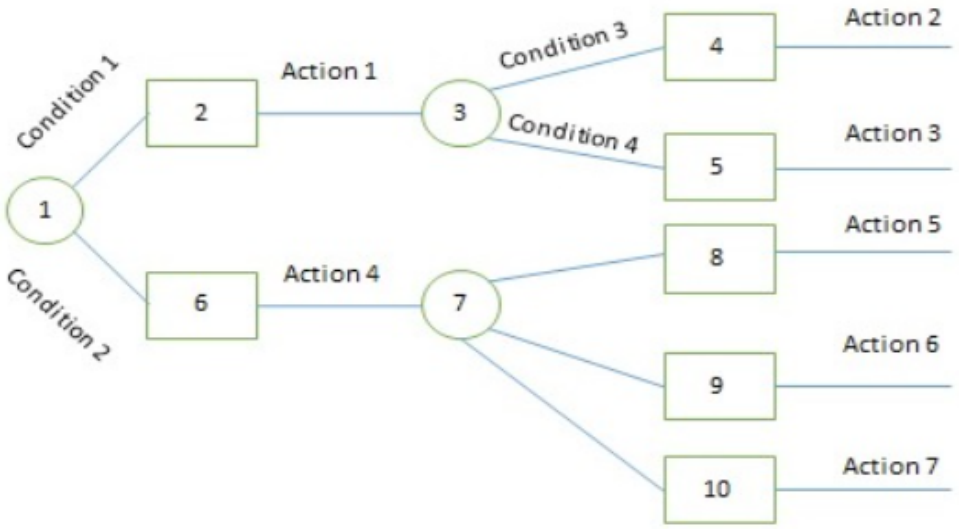
For more examples, check out your note copy

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Logic Modeling with Decision Trees

Decision trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication. A decision tree is a diagram that shows alternative actions and conditions within horizontal tree framework. Thus, it depicts which conditions to consider first, second, and so on.

Decision trees depict the relationship of each condition and their permissible actions. A square node indicates an action and a circle indicates a condition. It forces analysts to consider the sequence of decisions and identifies the actual decision that must be made.

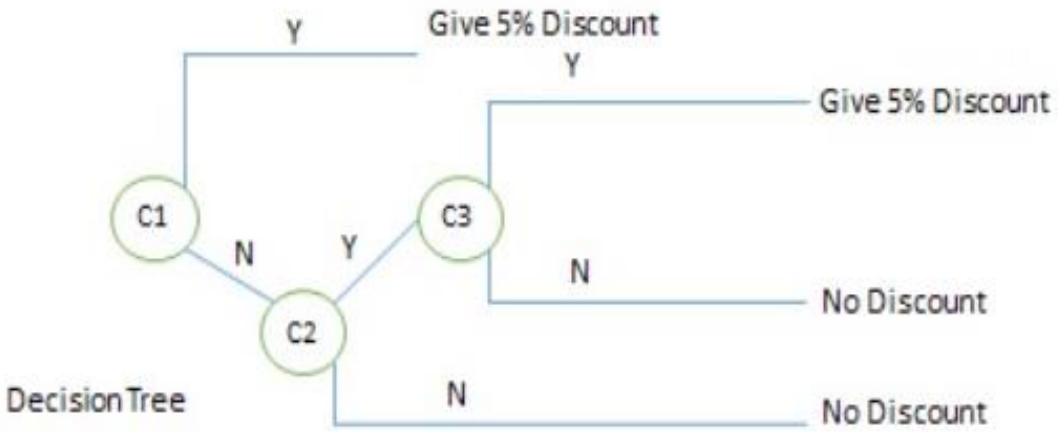


Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

The major limitation of a decision tree is that it lacks information in its format to describe what other combinations of conditions you can take for testing. It is a single representation of the relationships between conditions and actions.

For example, refer the following decision tree –



Y = Yes

N = No

C1 : Advance Payment Mode

C2 : Purchase Amount >= 10,000

C3 : Regular Customer

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

Contd. ….

For more examples, check out your note copy

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Conceptual Data Modeling

- The conceptual data model is a detailed model that captures the overall structure of data in an organization.
- It is independent DBMS or other implementation consideration.
- It is the representation of the organizational data.
- The purpose of a conceptual data model is to show as many rules about the meaning and interrelationship among data as possible.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Process of Conceptual Data Modeling

- The first step is used to develop a data model for the system being replaced.
- Next, a new conceptual data model is built that includes all the requirements of the new system.
- In the design stage, the conceptual data model is translated into a physical design, project repository links, all the design and data modeling steps performed during SDLC are also traced.
- **Primary deliverables (or outcomes) are the ER diagrams.**

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Gathering Information for Conceptual Data Modeling

- We typically do data modeling from a combination of the perspectives.
- The **first perspective** is generally called the **top-down approach**. This perspective defines the business rules for a data model from an understanding of the nature of the business. This perspective is typically the basis for a purchased data model.
- **Next perspective** is **bottom-up approach**. This perspective is obtained by reviewing specific business documents-computer-displays, reports, and business forms handled within the system.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Entity-Relationship Diagram (ERD)

- ER model is a detailed, logical and graphical representation of the entities, association and data elements for an organization or business area.
- An ER model is normally expressed as an ER diagram.
- An ERD illustrates the logical structure of databases. Thus, ERD is used to show the relationship among the entities, having their respective attributes in the form of the diagram.
- Peter Chen developed the concept of ERD in 1976 A.D.
- Thus, we can say that ERD is a pictorial representation showing the information formatted, stored and used by a business system.

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com

❑ Concepts/ Elements/ Design Issues of ERD

- Entity
- Attribute
- Composite vs. Simple (or atomic) attributes
- Single-valued vs. Multi-valued attributes
- Stored vs. Derived attributes
- Key attribute (or candidate key/ Primary key/ Identifier)
- Null-valued attribute
- Foreign key and Super key
- Weak Entity
- Relationship
- Relationship Degree
- Relationship Type (or Cardinality Ratio)
- Participation Constraints (Total and Partial)
- Examples of ERD

(Check out your note copy or PDF for these above mentioned topics and examples of designing ER-diagram).

Compiled by Ujjwal Rijal ||
rijalujjwal09@gmail.com